

PATENT OFFICE  
JAPANESE GOVERNMENT



This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: September 30, 1999

Application Number: Patent Application  
No. 11-279208

Applicant(s): CASIO COMPUTER CO., LTD.

April 7, 2000

Commissioner,  
Patent Office Takahiko Kondo

Certificate No. 2000-3024156

日 本 国 特 許 庁

PATENT OFFICE  
JAPANESE GOVERNMENT

JC925 U.S. P.  
09/670424  
09/26/00

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office.

出 願 年 月 日  
Date of Application:

1999年 9月30日

出 願 番 号  
Application Number:

平成11年特許願第279208号

出 願 人  
Applicant(s):

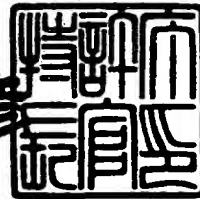
カシオ計算機株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2000年 4月 7日

特許庁長官  
Commissioner,  
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3024156

【書類名】 特許願

【整理番号】 A009904569

【提出日】 平成11年 9月30日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 12/00  
G09C 1/00

【発明の名称】 データベース管理装置、データベースシステム、暗号化装置及び記録媒体

【請求項の数】 9

【発明者】

【住所又は居所】 東京都羽村市栄町3丁目2番1号 カシオ計算機株式会社  
社羽村技術センター内

【氏名】 佐藤 誠

【発明者】

【住所又は居所】 東京都羽村市栄町3丁目2番1号 カシオ計算機株式会社  
社羽村技術センター内

【氏名】 竹田 恒治

【特許出願人】

【識別番号】 000001443

【氏名又は名称】 カシオ計算機株式会社

【代理人】

【識別番号】 100058479

【弁理士】

【氏名又は名称】 鈴江 武彦

【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9005919

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データベース管理装置、データベースシステム、暗号化装置及び記録媒体

【特許請求の範囲】

【請求項 1】 データベースの所定の列項目のデータについて当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化する暗号化手段と、

この暗号化手段によって暗号化されたデータベースを記憶する記憶手段とを具備したことを特徴とするデータベース管理装置。

【請求項 2】 前記所定の列項目を検索対象とした場合に、検索用として入力されたデータを前記所定の列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データと前記記憶手段に記憶された暗号化データベースの各項目データとを比較して検索処理を行うデータベース検索手段を具備したことを特徴とする請求項 1 記載のデータベース管理装置。

【請求項 3】 前記暗号化手段は、データベースの所定の列項目のデータを当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵と当該列項目に共通の列鍵とを複合的に用いて暗号化することを特徴とする請求項 1 記載のデータベース管理装置。

【請求項 4】 前記暗号化手段は、所定の関数に基づいて多次元空間に逐次ベクトルを発生させ、これらのベクトル成分を暗号のキーストリームとした暗号化方式を用い、前記行鍵および列鍵を前記関数の定数としてデータベースの暗号化を行うことを特徴とする請求項 1 記載のデータベース管理装置。

【請求項 5】 データベースを有する第 1 の情報端末と、この第 1 の情報端末にデータベースの検索を依頼する第 2 の端末とを有し、これらをネットワークを介して接続してなるデータベースシステムにおいて、

前記第 1 の情報端末側で、前記データベースの所定の列項目のデータについて当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化し、

前記第 2 の情報端末から前記所定の列項目を検索対象としてデータベースの検

索を依頼する際に、検索用として入力されたデータを当該列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データを前記ネットワークを介して前記第 1 の情報端末に送り、

前記第 1 の情報端末側で、前記検索用データに基づいて前記暗号化データベースの検索処理を行い、その検索結果として得られたデータを暗号化された状態で前記ネットワークを介して前記第 2 の情報端末に返信することを特徴とするデータベースシステム。

【請求項 6】 所定の列項目のデータが当該列項目に共通の列鍵を用いて暗号化されているデータベースを管理するデータベース管理装置であって、

所定の列項目を対象にしたデータ検索をする際、検索用に入力されたデータを前記列鍵を用いて暗号化する暗号化手段と、

この暗号化された検索用データと前記暗号化されているデータベースの各項目データとを比較して検索処理を行う検索手段と

を具備したことを特徴とするデータベース管理装置。

【請求項 7】 データベースの少なくとも 1 つの列項目データを当該列項目に共通の列鍵を用いて暗号化する暗号化装置であって、

暗号化の対象となる原データを取得する原データ取得手段と、

少なくとも前記列鍵によって決定される関数を用いて、 $n$  ( $n \geq 1$ ) 次元の空間の閉領域内に定義されたベクトルを逐次的に生成するベクトル生成手段と、

前記原データ取得手段によって得られた原データと前記ベクトル生成手段によって生成されたベクトルの成分との論理演算をビット単位で行って暗号データを生成する論理演算手段と

を具備したことを特徴とする暗号化装置。

【請求項 8】 コンピュータに、

データベースの所定の列項目のデータについて当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化する暗号化機能と、

この暗号化機能の実行結果として得られた暗号化データベースに対するデータ検索を行う検索機能と

を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 9】 所定の列項目のデータが当該列項目に共通の列鍵を用いて暗号化されているデータベースを管理するためのコンピュータに、

所定の列項目を対象にしたデータ検索をする際、検索用に入力されたデータを前記列鍵を用いて暗号化する暗号化機能と、

この暗号化された検索用データと前記暗号化されているデータベースの各項目データとを比較して検索処理を行う検索機能と

を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データベースを暗号化して管理するデータベース管理装置と、暗号化データベースを有する検索装置をネットワーク上に配置したデータベースシステムと、データベースの暗号化を行う暗号化装置及び記録媒体に関する。

【0002】

【従来の技術】

データベース管理装置では、データベースのセキュリティを確保するため、管理対象として保有しているデータベースを暗号化して保存しておくことが求められる。

【0003】

ここでセキュリティを高めるためには、より複雑な暗号処理を施すことで実現することができるが、複雑な暗号処理は演算に要する時間も大きくなることは避けられない。

【0004】

データベースは膨大なデータを有し、データ検索は、それらのデータの中からある項目について条件に合致するものを選び出し、条件に合う項目データを含むレコード（行データ）を出力するという処理になる。従って、大量のデータを取

り扱うデータ検索システムにおいては演算時間の増加は処理効率を悪化させる原因となる。

【0005】

【発明が解決しようとする課題】

上述したように、機密性の高いデータを記憶しているデータベースにはセキュリティの要求があり、セキュリティを高めるための暗号化処理はデータベースの利用性を妨げるという問題がある。

【0006】

本発明は前記のような課題を解決するためになされたもので、データベースのセキュリティを確保すると共に、データの高速検索を可能にしたデータベース管理装置、データベースシステム、暗号化装置及び記録媒体を提供することを目的とする。

【0007】

【課題を解決するための手段】

本発明は、データベースを暗号化する際に、検索に利用される列項目のデータについては当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化するようにしたものである。

【0008】

具体的には、本発明のデータベース管理装置は、データベースの所定の列項目のデータについて当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化する暗号化手段と、この暗号化手段によって暗号化されたデータベースを記憶する記憶手段とで構成される。

【0009】

このような構成されれば、データベースを暗号化する際に、各行毎に鍵を異ならせることでセキュリティを高めることができ、検索時には、検索用として入力されたデータを前記所定の列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データと暗号化データベースの各項目データとを比較することで高速検索を実現することができる。



## 【0010】

また、検索に利用される列項目以外の列項目のデータについて、各行毎に固有の行鍵と当該列項目に共通の列鍵とを複合的に用いて暗号化するようにすれば、さらにセキュリティを強化することができる。

## 【0011】

また、暗号化方式として、所定の関数に基づいて多次元空間に逐次ベクトルを発生させ、これらのベクトル成分を暗号のキーストリームとした多次元空間回転方式を用いれば、複雑な演算処理を必要とせずにデータベースを暗号化することができるので、演算能力の低い情報処理装置にも適用可能となる。

## 【0012】

また、データベースの保管場所を離間させておき、別の情報端末からネットワークを介して検索を依頼するようなデータベースシステムを構築することも可能である。この場合、前記のようにデータベースの所定の列項目（検索に利用される列項目）のデータを当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を暗号化しておく。そして、別の情報端末からデータベースの検索を依頼する際に、検索用データを当該列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データをネットワークを介して送るようにする。この検索用データを受けることで、前記暗号化データベースの検索処理を行い、その検索結果として得られたデータを暗号化された状態でネットワークを介して前記情報端末に返信する。したがって、常に暗号化された状態でデータをやり取りできるので、データベースのセキュリティを確保することができる。

## 【0013】

## 【発明の実施の形態】

以下、図面を参照して本発明の実施形態を説明する。

## 【0014】

## （第1の実施形態）

図1は本発明の第1の実施形態に係るデータベース管理装置の構成を示す図である。本装置は、行と列からなるマトリクス形式のデータベースを有し、そのデ

ータベースを暗号化して管理する機能と共に、入力された検索用データを暗号化して、その暗号化された検索用データに基づいてデータベースを検索する機能を備えたものであって、例えば磁気ディスク等の記録媒体に記録されたプログラムを読み込み、このプログラムによって動作が制御されるコンピュータによって実現される。

## 【0015】

図1に示すように、本装置には、CPU 11、表示装置12、入力装置13、プログラム記憶装置14、鍵記憶装置15、データ記憶装置16が設けられている。

## 【0016】

CPU 11は、本装置全体の制御を行うものであり、プログラム記憶装置14に記憶されたプログラムを読み込み、そのプログラムに従って各種処理を実行する。本実施形態において、CPU 11は図2に示すようなデータベースの暗号化処理や、図3および図4に示すようなデータベースの検索処理を実行する。

## 【0017】

表示装置12は、データを表示するためのデバイスであり、例えばLCD (Liquid Crystal Display) やCRT (Cathode-ray tube) 等が用いられる。入力装置13は、データを入力するためのデバイスであり、例えばキーボード、マウス等が用いられる。

## 【0018】

プログラム記憶装置14は、例えばROMあるいはRAMなどで構成され、本装置に必要なプログラムを記憶する。本装置に必要なプログラムとしては、データベース管理プログラムや暗号化プログラム等がある。

## 【0019】

なお、プログラム記憶装置14は半導体メモリの他に磁氣的、光学的記録媒体で構成することができる。この記録媒体はCD-ROM等の可搬型の媒体やハードディスク等の固定的な媒体を含む。また、この記録媒体に格納するプログラムは、その一部若しくは全部をサーバやクライアントからネットワーク回線などの伝送媒体を介して伝送制御部から受信する構成にしてもよく、更に、前記記録媒

体はネットワーク上に構築されたサーバの記録媒体であってもよい。更に、前記プログラムをネットワーク回線などの伝送媒体を介してサーバーやクライアントへ伝送してこれらの機器にインストールするように構成してもよい。

#### 【0020】

鍵記憶装置15は、例えばRAMなどで構成され、データベースの暗号化に用いられる鍵（行鍵と列鍵）を記憶する。

#### 【0021】

データ記憶装置16は、本装置に必要な各種のデータを記憶するためのデバイスであり、例えばRAMあるいは磁気ディスク装置等の外部記憶装置で構成される。このデータ記憶装置16には、データベースを格納しておくためのデータベース格納エリア16a、データベースを暗号化する際にオペレータにより設定された情報（検索対象項目、非暗号化項目等）を格納しておくための暗号化設定情報格納エリア16b、データベースを検索する際にオペレータにより設定された情報（対象列項目、検索文字列等）を格納しておくための検索設定情報格納エリア16c、データベース検索時の比較文字列を格納しておくための比較文字列格納エリア16dなどが設けられている。

#### 【0022】

ここで、本装置の動作を説明する前に、本装置によって実現されるデータベースの暗号化方法について説明しておく。

#### 【0023】

データベースを暗号化する場合に、各行毎（レコード毎）に異なる鍵を用いて暗号化すると、鍵の解読は困難となり、セキュリティ性を高めることができる。しかし、データベースを検索する際に、暗号化されているデータを各行毎の鍵で復号化するか、または、検索用として入力されたデータ（キーワード）を各行毎の鍵で暗号化しなければならいため、検索結果を得るのに時間がかかることになる。一方、各列毎に異なる鍵を用いてデータベースを暗号化すると、検索対象となる列項目に対応した鍵のみを用いて検索用データを暗号化すれば良いのでデータベース検索を高速に行うことができる。しかし、同じ列の中で同一のデータがあると、暗号化結果も同じになってしまうため、そこから鍵を解読される可能性

が高くなる。

【0024】

そこで、本発明では、データベースを暗号化する際に、検索に頻繁に利用される列項目のデータについては列共通鍵で暗号化し、その他の列項目のデータについては各行毎に異なる鍵を与えて暗号化することを特徴とする。つまり、各行毎に鍵を異ならせることで、セキュリティを高めると共に、検索時には検索に利用される項目に入力されたデータを列鍵で暗号化してデータベース上の暗号化データと比較することで高速検索を可能とするものである。

【0025】

図5は第1の実施形態におけるデータベースの構成を説明するための図であり、図5(a)は暗号化前の状態、同図(b)は暗号化後の状態、同図(c)は復号化後の状態を示している。また、図6は第1の実施形態における列鍵と行鍵の構成を示す図である。

【0026】

図5(a)に示すように、本装置では、行と列からなるマトリクス形式のデータベースを備えている。ここでは、個人データをデータベース化した場合を示している。このデータベースは、1レコードが「number」, 「name」, 「state」, 「weight」, 「height」, 「age」, 「phone」の各項目で構成するようにしている。

【0027】

このようなデータベースに対し、列鍵と行鍵を用いて暗号化を行う。すなわち、検索に頻繁に利用される列項目を「name」, 「state」, 「age」とした場合に、これらの列項目の各行のデータについては、図6に示すように、「apple」, 「orange」, 「lemon」といった各列項目に共通の列鍵を用いて暗号化し、その他の列項目「weight」, 「height」, 「phone」の各行のデータについては、各行毎に固有の鍵を用いて暗号化する。

【0028】

なお、「number」の列は暗号化を行わないものとする。行鍵としては、

「tiger」, 「dog」, 「cat」, 「mouse」, 「elephant」, 「cow」, 「pig」, 「rabbit」, 「lion」といった鍵が用いられる。これらの鍵は所定の関数を用いて各列または各行毎に数学的に発生させることができる。

【0029】

図5(a)のデータベースを列鍵と行鍵を用いて暗号化した結果を図5(b)に示す。データ記憶装置16のデータベース格納エリア16aには、図5(b)に示すような状態でデータベースが保存されることになる。

【0030】

データベースを検索する場合には、検索に利用される列項目に対応した列鍵を用いて検索用データを暗号化してから検索処理を行う。例えば、「state」の中の「Florida」といったデータを検索する場合には、まず、検索用データとして入力された「Florida」を「state」の列鍵「apple」で暗号化して、「h\*/fDD」を得る。この「h\*/fDD」といったデータを「state」の列の各行から検索する。これにより、「number2」と「number8」に該当するデータが存在することがわかる。

【0031】

また、暗号化されたデータベースを元に戻す場合には、暗号化の時と同じ列鍵と行鍵を用いる。図5(b)のデータベースを暗号化の時と同じ列鍵と行鍵を用いて復号化すると、図5(c)に示すように元のデータを得ることができる。

【0032】

以下に、本装置の動作について説明する。

【0033】

ここでは、(a)データベースを暗号化する場合の処理と、(b)データベースを検索する場合の処理に分けて説明する。なお、このフローチャートで示す各機能を実現するプログラムはCPUが読み取り可能なプログラムコードの形態で前記プログラム記憶装置14の記録媒体に格納されている。また、このプログラムはプログラムコードの形態でネットワーク回線などの伝送媒体を介して伝送することもできる。

【0034】

(a) データベースを暗号化する場合

図2は本装置にて実行されるデータベース暗号化処理の動作を示すフローチャートである。今、データベースが暗号化されていない状態でデータ記憶装置16のデータベース格納エリア16aに記憶されているものとする。この状態が図5(a)である。

【0035】

まず、図示せぬデータベース暗号化設定画面にて、暗号化対象となるデータベースを指定する(ステップA11)。

【0036】

次に、そのデータベースに設けられた各列項目の中で検索に利用する列項目と、暗号化を必要としない列項目をそれぞれ設定する(ステップA12)。図5(a)の例では、検索に利用する列項目は「name」、「state」、「age」であり、暗号化を必要としない列項目は「number」である。ここでの設定情報はデータ記憶装置16の暗号化設定情報格納エリア16bに記憶される。

【0037】

次に、当該データベースの暗号化に用いる行鍵と列鍵を決定する(ステップA13)。ここで決定された行鍵と列鍵の情報は鍵記憶装置15に記憶される。

【0038】

このような設定操作の後、データベースの中の各列項目を順次指定していくと(ステップA14)、その指定された列項目に対する暗号化方式が前記設定情報に基づいて判断される(ステップA15)。この場合、データベースの中の「number」の列項目については非暗号化項目として設定されているので、何もしない。つまり、「number」の項目は元データのままである。

【0039】

また、指定された列項目が検索に利用する列項目として設定されている場合には、鍵記憶装置15に記憶された当該列項目に共通の列鍵が読み出され(ステップA15→A16)、当該列項目の各行のデータがその列鍵により暗号化される。

(ステップA17)。すなわち、データベースの「name」, 「state」, 「age」の項目の各行のデータは、図6に示すように、「apple」, 「orange」, 「lemon」といった各列に固有の鍵を用いて暗号化されることになる。

## 【0040】

また、指定された列項目が検索に利用する列項目として設定されていない場合、つまり、他の列項目であった場合には、鍵記憶装置15に記憶された各行に対応した行鍵が読み出され(ステップA15→A18)、当該列項目の各行のデータがそれぞれに固有の行鍵により暗号化される(ステップA19, A20)。すなわち、データベースの中の「state」, 「weight」, 「height」の各項目のデータに関しては、図6に示すように、1行目のデータは「tiger」, 2行目のデータは「dog」, 3行目のデータは「cat」, 4行目のデータは「mouse」, 5行目のデータは「elephant」, 6行目のデータは「cow」, 7行目のデータは「pig」, 8行目のデータは「rabbit」, 9行目のデータは「lion」といったように、各行に対応した行鍵を用いて暗号化される。

## 【0041】

このような暗号化処理がデータベースの各列項目について繰り返し行なわれる。すべての列項目の各行のデータの暗号化が終了すると、その暗号化データベースがデータ記憶装置16のデータベース格納エリア16aに上書き保存される(ステップA22)。この状態が図5(b)である。

## 【0042】

## (b) データベースを検索する場合

図3は本装置にて実行されるデータベース検索処理の動作を示すフローチャートである。今、前記(a)で説明した暗号化処理にて、データベースが暗号化されてデータ記憶装置16に保存されているものとする。

## 【0043】

まず、図3(a)のフローチャートに示すように、図示せぬデータベース検索用設定画面にて、検索情報の入力を行う(ステップB11)。検索情報の入力と

は、検索対象となる列項目と、検索用の文字列（キーワード）を入力することである。これらの入力情報はデータ記憶装置 16 の検索設定情報格納エリア 16 c に格納される。入力装置 13 を通じて検索情報が入力されると、検索前処理が実行される（ステップ B 12）。

## 【0044】

この検索前処理では、図 3（b）のフローチャートに示すように、検索対象として入力された列項目が所定の列項目であるか否かが判断され（ステップ C 11）、所定の列項目であることが判明した場合には（ステップ C 11 の Yes）、その列項目に共通の列鍵で検索用の文字列が暗号化される（ステップ C 12）。

## 【0045】

所定の列項目とは、前記データベースの暗号化時に設定された検索対象項目（検索に利用される項目）であり、具体的には「name」、「state」、「age」の各項目が該当する。この検索対象項目に関する情報はデータ記憶装置 16 の暗号化設定情報格納エリア 16 b に格納されている。したがって、前記ステップ C 11 では、この暗号化設定情報格納エリア 16 b を参照して所定の列項目であるか否かの判断を行うことになる。また、列項目に共通の列鍵は鍵記憶装置 15 に格納されている。したがって、前記ステップ C 12 では、この鍵記憶装置 15 から当該列項目に対応した列鍵を読み出して検索用の文字列を暗号化することになる。例えば、指定項目が「state」であれば、「orang」といった列鍵を用いて検索用の文字列を暗号化することになる。

## 【0046】

また、検索対象として入力された列項目が所定の列項目でなかった場合には（ステップ C 11 の No）、前記のような検索用文字列の暗号化は行われず。

## 【0047】

このような検索前処理の後、データベースの検索処理（図 4 参照）が行われ（ステップ B 13）、その検索結果として得られたデータが表示装置 12 に表示される（ステップ B 14）。図 4 にデータベースの検索処理を示す。

## 【0048】

図 4 は前記ステップ B 13 の検索処理の動作を具体的に示すフローチャートで



ある。

【0049】

まず、図4（a）のフローチャートに示すように、検索用文字列がデータベースとの比較文字列としてデータ記憶装置16の比較文字列格納エリア16dにセットされる（ステップD11）。この場合、上述したように検索対象として入力された列項目が所定の列項目（「name」、「state」、「age」）であった場合には、前記検索前処理によって、当該検索用文字列がその列項目に対応した列鍵にて暗号化されて比較文字列格納エリア16dにセットされる。その他の列項目の場合には、暗号化されることなく、そのままの状態と比較文字列格納エリア16dにセットされる。

【0050】

次に、データ記憶装置16のデータベース格納エリア16aに格納された暗号化データベースの列番号による暗号化方式が判断される（ステップD12）。これにより、検索対象が列鍵で暗号化された所定の列項目に該当する場合には、その対象列の各行にあるデータが順次走査され（ステップD12→D13）、指定された行に含まれた対象項目のデータの文字列と前記比較文字列格納エリア16dにセットされた検索用文字列（暗号化された文字列）との比較処理が行われる（ステップD14）。

【0051】

この比較処理では、図4（b）のフローチャートに示すように、データベースから取り出された対象項目のデータの暗号化文字列と検索用の暗号化文字列とを比較して、両者が一致するか否かを判断する（ステップE11）。両者が一致した場合には（ステップE11のYes）、その一致した項目を含むレコードデータをデータベース検索結果として抽出する（ステップE12）。

【0052】

この処理を暗号化データベースの終端まで繰り返して、該当するデータを順次抽出し（ステップD15）、この抽出したデータを検索結果として出力する（ステップD20）。

【0053】

具体的に説明すると、図5（b）の暗号化データベースの例で、例えば「state」の項目の中の「Florida」といったデータを検索することが指定された場合には、まず、検索用データとして入力された「Florida」を「state」の列鍵「apple」で暗号化して、「h\*/fDD」を得る。この「h\*/fDD」といったデータを「state」の列から検索する。これにより、「number2」と「number8」に該当するデータが存在することがわかる。

## 【0054】

一方、検索対象が行鍵で暗号化されたその他の列項目に該当する場合には、その対象列の各行にあるデータが順次走査され（ステップD12→D16）、指定された行に含まれた対象項目のデータが各行に固有の行鍵によって復号化された後（ステップD17）、前記比較文字列格納エリア16dにセットされた検索用文字列（非暗号化文字列）との比較処理が行われる（ステップD18）。

## 【0055】

この比較処理では、図4（b）のフローチャートに示すように、データベースから取り出された対象列のデータの復号化文字列と検索用の非暗号化文字列との比較により、両者が一致するか否かを判断する（ステップE11）。両者が一致した場合には（ステップE11のYes）、その一致した項目を含むレコードデータをデータベース検索結果として抽出する（ステップE12）。

## 【0056】

この処理を暗号化データベースの終端まで繰り返して、該当するデータを順次抽出し（ステップD19）、この抽出したデータを検索結果として出力する（ステップD20）。

## 【0057】

具体的に説明すると、図5（b）の暗号化データベースの例で、例えば「weight」の項目の中の「163」といったデータを検索することが指定されたとした場合に、まず、「weight」の1行目のデータを「tiger」といった行鍵で復号化する。同様に、2行目のデータを「dog」、3行目のデータを「cat」、4行目のデータを「mouse」、5行目のデータを「e1

「e p h a n t」, 6行目のデータを「c o w」, 7行目のデータを「p i g」, 8行目のデータを「r a b b i t」, 9行目のデータを「l i o n」といったように、各行に対応した行鍵を用いて、それぞれ復号化した後に、検索用データとして入力された「1 6 3」に基づいて「s t a t e」の列から該当するデータを検索する。これにより、「n u m b e r 3」と「n u m b e r 9」に該当するデータが存在することがわかる。

【0058】

このように、データベースを暗号化する際に、検索に利用される所定の列項目については列共通鍵で暗号化し、検索時には、検索用データをその列共通鍵で暗号化してデータベース上の暗号化データと比較することで高速検索を実現できる。また、所定の列項目以外の列項目については各行毎に異なる鍵を与えて暗号化することでセキュリティを高める。この場合、検索時には各行毎の鍵を用いた復号化を必要とするため、前記所定の列項目に対する検索に比べて時間はかかるものの、検索に頻繁に利用される項目ではないので問題にはならない。

【0059】

(第2の実施形態)

前記第1の実施形態では、所定の列項目以外の列項目のデータについては、各行毎に固有の行鍵を用いて暗号化するようにしたが、第2の実施形態では、さらにセキュリティを高めるため、各行毎に固有の行鍵と当該列項目に共通の列鍵とを複合的に用いて暗号化することを特徴とする。

【0060】

図7は第2の実施形態におけるデータベースの構成を示す図であり、図7(a)は暗号化前の状態、同図(b)は暗号化後の状態、同図(c)は復号化後の状態を示している。また、図8は第2の実施形態における合成鍵の構成を示す図である。

【0061】

図7(a)に示すように、本装置では、行と列からなるマトリクス形式のデータベースを備えている。ここでは、個人データをデータベース化した場合を示している。このデータベースは、1レコードが「n u m b e r」, 「n a m e」,

「state」, 「weight」, 「height」, 「age」, 「phone」の各項目で構成するようにしている。

【0062】

このようなデータベースに対し、合成鍵を用いて暗号化を行う。すなわち、検索に頻繁に利用される列項目を「name」, 「state」, 「age」とした場合に、これらの列項目の各行のデータについては、図8に示すように、「apple」, 「orange」, 「lemon」といった各列項目に共通の列鍵を用いて暗号化し、その他の列項目「weight」, 「height」, 「phone」の各行のデータについては、「banana+行鍵」, 「lychee+行鍵」, 「apricot+行鍵」といったように、列鍵と行鍵とを組み合わせさせて暗号化する。

【0063】

なお、「number」の列は暗号化を行わないものとする。行鍵としては、「tiger」, 「dog」, 「cat」, 「mouse」, 「elephant」, 「cow」, 「pig」, 「rabbit」, 「lion」といった鍵が用いられる。これらの鍵は所定の関数を用いて各列または各行毎に数学的に発生させることができる。

【0064】

図7(a)のデータベースを合成鍵を用いて暗号化した結果を図7(b)に示す。データ記憶装置16のデータベース格納エリア16aには、図7(b)に示すような状態でデータベースが保存されることになる。

【0065】

データベースを検索する場合に、前記第1の実施形態と同様に、検索に利用される列項目に共通の列鍵を用いて検索用データを暗号化してから検索処理を行えば良い。例えば、「state」の中の「Florida」といったデータを検索する場合には、まず、検索用データとして入力された「Florida」を「state」の列鍵「apple」で暗号化して、「h\*/fDD」を得る。この「h\*/fDD」といったデータを「state」の列の各行から検索する。これにより、「number2」と「number8」に該当するデータが存在

することがわかる。

【0066】

また、暗号化されたデータベースを元に戻す場合には、暗号化の時と同じ合成鍵を用いる。図7(b)のデータベースを暗号化の時と同じ合成鍵を用いて復号化すると、図7(c)に示すように元のデータを得ることができる。

【0067】

なお、データベースを暗号化する場合の処理や、暗号化されたデータベースを検索する場合の処理については、所定の列項目以外の列項目の各行のデータに対して列鍵と行鍵を組み合わせて暗号化する点を除いては前記第1の実施形態の処理(図2～図4)と同様であるため、ここでは、その説明を省略するものとする。

【0068】

このように、検索に頻繁に利用される列項目については、その列項目に共通の列鍵を用いて暗号化することで、前記第1の実施形態と同様に高速検索を実現できるものであり、また、他の列項目については、列鍵と行鍵とを複合的に用いて暗号化することで、さらにセキュリティを強化することができる。

【0069】

(第3の実施形態)

前記第1または第2の実施形態では、本発明を装置単体で構成したが、データベースの保管場所を離間させておき、別の情報端末からネットワークを介して検索を依頼するようなデータベースシステムを構築することも可能である。

【0070】

以下に、このようなデータベースシステムについて説明する。

【0071】

図9は本発明の第3の実施形態に係るデータベースシステムの構成を示すブロック図である。本システムは、第1の端末装置20と第2の端末装置30とを有する。第1の端末装置20と第2の端末装置30とはネットワーク40を介して接続されている。

【0072】

第 1 の端末装置 2 0 は、データベースサービスを行うサーバコンピュータとして用いられるものであって、データベースの検索処理を行う検索装置 2 1 と、データベースを保存しておくためのデータ記憶装置 2 2 とで構成される。第 2 の端末装置 3 0 は、データベースの検索を第 1 の端末装置 2 0 に依頼し、その結果を第 1 の端末装置 2 0 から受け取るクライアントコンピュータとして用いられるものであって、検索依頼装置 3 1 と復号化装置 3 2 とで構成される。

## 【 0 0 7 3 】

このようなデータベースシステムにあつては、第 1 の端末装置 2 0 において、図 2 で説明したように、データベースの所定の列項目の各行のデータを当該列項目に共通の列鍵を用いて暗号化し、その他の列項目の各行のデータについては各行毎に固有の行鍵を用いて暗号化してデータ記憶装置 2 2 に保存しておくものとする。

## 【 0 0 7 4 】

ここで、第 2 の端末装置 3 0 から第 1 の端末装置 2 0 に対してデータベースの検索を依頼する際に、図 3 の検索前処理までを第 2 の端末装置 3 0 側で実行する。すなわち、第 2 の端末装置 3 0 の検索依頼装置 3 1 によって、検索対象として入力された列項目が所定の列項目であるか否かを判断し、所定の列項目である場合に、検索用の文字列（キーワード）を当該列項目に共通の列鍵を用いて暗号化する。その他の列項目が検索対象である場合には、このような暗号化は不要である。

## 【 0 0 7 5 】

検索前処理の後、第 2 の端末装置 3 0 から検索用文字列をネットワーク 4 0 を介して第 1 の端末装置 2 0 に送る。第 1 の端末装置 2 0 側では、この検索用文字列を受けることにより、図 4 で説明したような検索処理を実行する。

## 【 0 0 7 6 】

すなわち、第 1 の端末装置 2 0 の検索装置 2 1 により、検索対象が所定の列項目か否かを判断し、所定の列項目であれば、第 2 の端末装置 3 0 から取得した検索用文字列（暗号化文字列）とデータ記憶装置 2 2 内の暗号化データベースの当該列項目の各行のデータとを比較して、該当するデータを抽出するといった処理

を行う。また、検索対象が所定の列項目以外の列項目であれば、データ記憶装置 2 2 内の暗号化データベースの当該列項目のデータを各行毎の鍵で復号化してから、第 2 の端末装置 3 0 から取得した検索用文字列（非暗号化文字列）とその復号化された各行のデータとを比較して、該当するデータを抽出するといった処理を行う。

## 【 0 0 7 7 】

このようにして検索結果が得られると、第 1 の端末装置 2 0 はその検索結果として得られたデータを暗号化の状態のままでネットワーク 4 0 を介して第 2 の端末装置 3 0 に返信する。第 2 の端末装置 3 0 は、第 1 の端末装置 2 0 と共通の暗号鍵を持っている。したがって、第 1 の端末装置 2 0 から検索結果を受信すると、内部の復号化装置 3 2 にて、そのデータを暗号鍵を用いて復号化することができる。この場合、第 1 の端末装置 2 0 と第 2 の端末装置 3 0 との間で、常に暗号化された状態でデータがやり取りされるので、データベースのセキュリティを確保することができる。

## 【 0 0 7 8 】

このように、第 1 の端末装置 2 0 側にデータベースを持たせて、第 2 の端末装置 3 0 からのアクセスによってデータベース検索を行うようにしたデータベースシステムであっても、検索に頻繁に使う列項目のデータを当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化しておくことで、セキュリティを高めると共に高速検索を実現することができる。

## 【 0 0 7 9 】

なお、前記所定の列項目以外の列項目のデータについては、前記第 2 の実施形態で説明したように、各行毎に固有の行鍵と当該列項目に共通の列鍵とを複合的に用いて暗号化することでも良く、このようにすれば、さらにセキュリティを強化することができる。

## 【 0 0 8 0 】

（暗号化方式）

次に、本発明に適用されるデータベースの暗号化方式について説明する。

## 【0081】

本発明では、データベースの暗号化アルゴリズムとして多次元空間回転方式（多次元空間ベクトル方式）を用いる。この多次元空間回転方式は、所定の関数に基づいて多次元空間に逐次ベクトルを発生させ、これらのベクトル成分を暗号のキーストリームとするものである。この多次元空間回転方式では、演算能力の低い情報処理装置でも計算できるので、携帯型の端末に適用するのに相応しい。つまり、本発明のデータベースを外部からアクセスするような環境において、データの機密性を保持して処理するためには、このような暗号化方式を採用することが望ましい。

## 【0082】

以下に、図10および図11を参照して、多次元空間回転方式を用いた暗号化と復号化の処理を示す。

## 【0083】

図10（a）は多次元空間回転方式を用いた場合での暗号化の処理動作を示すフローチャートである。

## 【0084】

暗号化の対象となる原データをMとする（ステップF11）。このデータMは、バイナリデータである。まず、このデータMに対してbit単位でスクランブル1をかける（ステップF12）。こうして得られたデータをM'とする（ステップF13）。

## 【0085】

ここで、このデータM'に数学的に順次生成される乱数をXOR（排他的論理和）して暗号化を行う（ステップF14）。このとき、乱数の発生関数に多次元ベクトルrを用いることが多次元空間回転方式の特徴である。この場合、多次元ベクトルrを発生させる関数、あるいは、その関数に用いられる定数は暗号化キー（秘密・公開キー）で決定される。

## 【0086】

すなわち、暗号化に際し、秘密キー（ $P_1$ ， $P_2$ ）および公開キー（ $P_3$ ， $P_4$ ）を定数として用いた関数に従って多次元ベクトルrを生成し、 $M' \text{ XOR } r$ と



いった論理演算を行ってデータM'を暗号化する。こうして得られた暗号データをCとする（ステップF15）。

【0087】

具体的に説明すると、例えば図11に示すように、 $r$ が3次元ベクトル $(x, y, z)$ であり、そのベクトル成分 $x, y, z$ の計算精度が16bitであるとする。この3次元ベクトル $r(x, y, z)$ を後述する式(1)に従って $r_0, r_1, r_2, r_3 \dots$ といったように順次発生する。

【0088】

今、データMが8bitデータの並びとして、 $m_0 m_1 m_2 m_3 m_4 m_5 m_6 \dots$ と与えられたとき、Mは前記計算精度(16bit)に基づいて、その要素の2つ(8bit)ずつに分解される。そして、3次元ベクトルが $r_0$ である場合には、データMと $r_0(x_0, y_0, z_0)$ とのXOR(排他的論理和)により、 $(x \text{ XOR } m_0 m_1) (y \text{ XOR } m_2 m_3) (z \text{ XOR } m_4 m_5) \dots$ といった計算が行われ、この計算結果として、 $C_0 C_1 C_2 C_3 C_4 C_5 \dots$ といった暗号データCが得られる。

【0089】

このようにして得られたデータCに対して、さらにbit単位でスクランブル2をかける（ステップF16）。こうして得られたデータをC'とし、最終的な暗号データとして出力する（ステップF17）。

【0090】

なお、前記の処理で、C'を改めてM'と見なして前記同様の暗号化を繰り返すことによって、復号化の困難さのレベルを上げることができる。また、このときの多次元ベクトル $r$ を生成する関数の形を変えれば、復号化の困難さはさらに増すことになる。

【0091】

次に、復号化の処理動作について説明する。

【0092】

図6(b)は復号化の処理動作を示すフローチャートである。

【0093】

復号化は、基本的には前記暗号化処理の逆の処理を行えば良い。

【0094】

すなわち、暗号データを  $C'$  とすると（ステップG11）、まず、このデータ  $C$  に対して  $\text{bit}$  単位で前記暗号化時に行ったスクランブル2とは逆のスクランブル2をかける（ステップG12）。これにより、スクランブル2をかける前のデータ  $C$  が得られる（ステップG13）。

【0095】

次に、 $C \text{ XOR } r$  といった計算を行ってデータ  $C$  を復号化する（ステップG14）。これにより、暗号化を行う前のデータ  $M'$  が得られる（ステップG15）。

【0096】

そして、このデータ  $M'$  に対しても  $\text{bit}$  単位で前記暗号化時に行ったスクランブル1とは逆のスクランブル1をかける（ステップG12）。これにより、スクランブル1をかける前のデータつまり原データ  $M$  が得られる（ステップG17）。

【0097】

なお、暗号化の処理で、 $C'$  を改めて  $M'$  と見なして暗号化を繰り返したり、多次元ベクトル  $r$  を生成する関数の形を変えたりする処理が加わっていれば、その処理に対応させて復号化の処理を行うものとする。

【0098】

ところで、多次元ベクトル  $r$  を用いた暗号化において、多次元ベクトル  $r$  を扱う関数を決定するパラメータ（定数）の集合  $P$  を2つの部分に分けて、

$$P = \{P_s, P_p\}$$

と表わす。ここで、 $P_s$  は秘密パラメータであり、暗号化キー（秘密キー  $P_1$  ,  $P_2$  ）に相当する。 $P_p$  は公開パラメータであり、暗号化キー（公開キー  $P_3$  ,  $P_4$  ）に相当する。

【0099】

次に、多次元空間回転方式について、さらに詳しく説明する。

【0100】

$n$  ( $n \geq 1$ ) 次元の空間を張るベクトルを  $r$  とし、その初期値  $r_0$  から逐次的に新しいベクトル  $r_i$  ( $i = 0, 1, 2, 3 \dots$ ) を発生する関数を  $R$  とする。このとき、 $r_i$  は以下のような式 (1) で表わされる。

【0 1 0 1】

$$r_i = A \cdot R(P, r_{i-1}) r_{i-1} + c \quad \dots (1)$$

ここで、 $A$  は適当な定数係数である。 $P$  は関数に使用される定数の集合であり、秘密キー  $P_1, P_2$  と、公開キー  $P_3, P_4$  が使用される。 $c$  はベクトルを並進移動する定数ベクトルである。

【0 1 0 2】

前記式 (1) において、係数  $A$  は関数  $R$  に適当な制約 (例えば  $|R| \leq 1$ ) を設けたときに、各ベクトルが多次元空間の閉空間領域内に存在するための条件を与える。定数ベクトル  $c$  はベクトル  $r_i$  が収斂するとき、「トリビアル」な点 (例えば  $r = 0$  のような無意味な点) にならないことを保証する (勿論  $c = 0$  も許される)。

【0 1 0 3】

$n$  次元空間では、ベクトル  $r$  は  $n$  個の要素を持つ。 $r = (x_1, x_2 \dots x_n)$ 。計算機上では、数値データ一般に、コンパイラが定義するビット長 ( $m$ ) の精度 (例えば 8 バイト又は 64 ビット) で表現されている。したがって、ベクトルの逐次生成法のある瞬間に  $n \times m$  のデータの精度でベクトル  $r$  を再現できないと、それに続くベクトル  $r$  は正確には再現できない (あるいはそうなるような関数  $R$  を定義する)。これは、ベクトル  $r$  の初期値  $r_0$  についても同様で、初期値  $r_0$  を  $n \times m$  のデータの精度で再現したときにのみ、それに続くベクトル  $r_1, r_2, r_3 \dots$  の再現性が保証される。

【0 1 0 4】

前記式 (1) を用いて得られたベクトル  $r$  の成分を、そのデータ定義長に応じて 1 個乃至複数個並べて、その総ビット長に対応した文字列 (1 文字あたり 8 ビットが普通) とビット毎の排他論理演算 (XOR) を行う。これを第 1 暗号化とする。これについては、前記図 1 1 で述べた通りある。

【0 1 0 5】

この手続きは暗号解読への対策となる。この場合、再度式(1)、関数Rを変えて新しいベクトルを発生させ、第1暗号化と同じ手法により暗号化を行っても良い。これを第2暗号化とする。

【0106】

具体的な例として、 $n=2$ の場合を考える。

【0107】

まず、 $r_{i-1}$  をこの平面に立てた法線の周りに $\theta$ だけが回転する演算として、Rを定義すれば、Rは $2 \times 2$ のマトリックスとなり、次のように表わされる。

【0108】

【数1】

$$R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad \dots (2)$$

【0109】

この場合、 $\theta$ は一種の媒介変数となっている。すなわち、媒介変数が $r_{i-1}$ の関数として与えられ、

$$\theta(r) = f(P, r) \quad \dots (3)$$

で表わされるとき、前記式(2)で表わされる変換は、形式的に前記式(1)で表わされる。

【0110】

なお、前記式(3)において、Pは関数fに使われる定数の集合として定義されるものであり、秘密キー $P_1$ 、 $P_2$ と、公開キー $P_3$ 、 $P_4$ が使用される。

【0111】

このように、逐次的に発生させた多次元空間を張るベクトルrを暗号化に使うことで、RSAのような暗号化に比べ、計算機の処理精度や処理能力に依存しない暗号化を実現できる。

【0112】

さて、このような多次元空間回転方式を本発明のデータベースの暗号化に適用する場合には、行鍵と列鍵を

$$r' = A \cdot R(P, r) r + c \quad \dots (4)$$

の関数RのPとして利用する。前記式(4)のPは関数Rに使用される定数の集合である。本発明の場合には、「apple」、「orange」、「lemon」等を秘密キーとして用いる。すなわち、前記の説明では、秘密キーとして $P_1$ 、 $P_2$ と定めているので、例えば「apple」を秘密キーとする場合には、 $P_1$ を「ap」、 $P_2$ を「ple」とする。なお、これは一例であり、鍵の設定の仕方はこれに限らず、いろいろな組み合わせが考えられる。

【0113】

このように、多次元空間回転方式を本発明のデータベースの暗号化に適用する場合には、行鍵と列鍵を多次元空間回転方式の所定の関数の定数成分として用いる。これにより、高精度の演算能力を必要とせずに、解読困難な暗号結果を得ることができる。

【0114】

なお、多次元空間回転方式では、次元数が大きくなると、回転行列R(関数)の要素も多くなり、暗号化/復号化時の演算負荷が大きくなる問題がある。このような問題を解消する方法として、多次元空間回転方式を用いた暗号化方式における多次元空間回転行列を次数の少ない疑似空間回転行列に置き換えて計算する方法がある。

【0115】

具体的に説明すると、例えば、6次元空間における回転行列Rの要素は、式(5)のようになり、計算量が膨大に増える。

【0116】

【数2】

$$R = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} & R_{1,4} & R_{1,5} & R_{1,6} \\ R_{2,1} & R_{2,2} & R_{2,3} & R_{2,4} & R_{2,5} & R_{2,6} \\ R_{3,1} & R_{3,2} & R_{3,3} & R_{3,4} & R_{3,5} & R_{3,6} \\ R_{4,1} & R_{4,2} & R_{4,3} & R_{4,4} & R_{4,5} & R_{4,6} \\ R_{5,1} & R_{5,2} & R_{5,3} & R_{5,4} & R_{5,5} & R_{5,6} \\ R_{6,1} & R_{6,2} & R_{6,3} & R_{6,4} & R_{6,5} & R_{6,6} \end{bmatrix} \quad \dots (5)$$

【0117】

そこで、このような回転行列Rを疑似回転行列Qに置き換えて計算する。疑似

回転行列Qは、次数の少ない空間回転行列の要素を対角に配置し、残りの要素を0とした擬似的な行列である。例えば、6次元空間の場合には、式(6)のような疑似回転行列Qを用いる。

【0 1 1 8】

【数3】

$$Q = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & 0 & 0 & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & 0 & 0 \\ A_{3,1} & A_{3,2} & A_{3,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{1,1} & B_{1,2} & B_{1,3} \\ 0 & 0 & 0 & B_{2,1} & B_{2,2} & B_{2,3} \\ 0 & 0 & 0 & B_{3,1} & B_{3,2} & B_{3,3} \end{bmatrix} \quad \dots (6)$$

【0 1 1 9】

なお、式(6)において、A、Bは3次元回転行列である。

【0 1 2 0】

この疑似回転行列Qと前記回転行列Rの要素を比較して見ると、Qの要素には0が多く、計算量が少なくて済む。しかも、暗号化としての機能も十分に果たす。一般的には、式(7)のように、多次元空間回転行列Qを設定すれば良い。

【0 1 2 1】

【数4】

$$Q = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_i \end{bmatrix} \quad \dots (7)$$

( $A_1, A_2, \dots, A_i$  は多次元空間回転行列)

【0 1 2 2】

このように、多次元空間回転方式における回転行列を次元数の少ない複数の回転行列の要素を対角に配置し、残りを0要素とした疑似回転行列に置き換えて計算することで、計算量を大幅に減らして速やかに暗号化または復号化の実行することができる。

【0 1 2 3】

また、別の方法として、

$$P = S \cdot Q \cdot S^T \quad \dots (8)$$

として得られるPを新たな疑似空間回転行列として用いても良い。

【0124】

式(8)において、Qは前記疑似回転行列である。Sは置換行列であり、式(9)のように、各行各列に1の要素が1つ含まれた正方行列である。

【0125】

【数5】

$$S = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \dots (9)$$

【0126】

例えば、疑似回転行列Qが前記式(6)で表現される場合(6次元の場合)には、疑似空間回転行列Pは、式(10)のように表せる。

【0127】

【数6】

$$P = \begin{bmatrix} A_{1,1} & 0 & A_{1,2} & 0 & A_{1,3} & 0 \\ 0 & B_{1,1} & 0 & B_{1,2} & 0 & B_{1,3} \\ A_{2,1} & 0 & A_{2,2} & 0 & A_{2,3} & 0 \\ 0 & B_{2,1} & 0 & B_{2,2} & 0 & B_{2,3} \\ A_{3,1} & 0 & A_{3,2} & 0 & A_{3,3} & 0 \\ 0 & B_{3,1} & 0 & B_{3,2} & 0 & B_{3,3} \end{bmatrix} \quad \dots (10)$$

【0128】

これは、置換行列Sが式(11)のような場合である。

【0129】

【数7】

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \dots (11)$$

【0 1 3 0】

このように、多次元空間回転方式における回転行列を次元数の少ない複数の回転行列の要素を対角に配置し、残りを 0 要素とした疑似回転行列に置換行列を組み合わせてできる新たな疑似回転行列に置き換えて計算すれば、計算の行程が複雑化するため、暗号の解読をさらに困難なものにできる。

【0 1 3 1】

【発明の効果】

以上詳記したように本発明によれば、データベースを暗号化する際に、検索に利用される列項目のデータについては当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては各行毎に固有の行鍵を用いて暗号化するようにしたため、各行毎に鍵を異ならせることでセキュリティを高めることができ、検索時には、検索用として入力されたデータを前記所定の列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データと暗号化データベースとを比較することで高速検索を実現することができる。

【図面の簡単な説明】

【図 1】

本発明の第 1 の実施形態に係るデータベース管理装置の構成を示す図。

【図 2】

前記データベース管理装置にて実行されるデータベース暗号化処理の動作を示すフローチャート。

【図 3】

前記データベース管理装置にて実行されるデータベース検索処理の動作を示すフローチャート。

【図 4】

前記図 3 のステップ B 1 3 の検索処理の動作を具体的に示すフローチャート。

【図 5】

第 1 の実施形態におけるデータベースの構成を説明するための図であり、図 5 (a) は暗号化前の状態、同図 (b) は暗号化後の状態、同図 (c) は復号化後の状態を示す図。



【図 6】

第 1 の実施形態における列鍵と行鍵の構成を示す図。

【図 7】

本発明の第 2 の実施形態におけるデータベースの構成を示す図であり、図 7 (a) は暗号化前の状態、同図 (b) は暗号化後の状態、同図 (c) は復号化後の状態を示す図。

【図 8】

第 2 の実施形態における合成鍵の構成を示す図。

【図 9】

本発明の第 3 の実施形態に係るデータベースシステムの構成を示すブロック図。

【図 1 0】

多次元空間回転方式を用いた場合での暗号化処理と復号化処理の動作を示すフローチャート。

【図 1 1】

多次元空間回転方式を用いた場合での暗号化の演算方法を説明するための図。

【符号の説明】

- 1 1 … CPU
- 1 2 … 表示装置
- 1 3 … 入力装置
- 1 4 … プログラム記憶装置
- 1 5 … 鍵記憶装置
- 1 6 … データ記憶装置
- 1 6 a … データベース格納エリア
- 1 6 b … 暗号化設定情報格納エリア
- 1 6 c … 検索設定情報格納エリア
- 1 6 d … 比較文字列格納エリア
- 2 0 … 第 1 の端末装置
- 2 1 … 検索装置

2 2 …データ記憶装置

3 0 …第 2 の端末装置

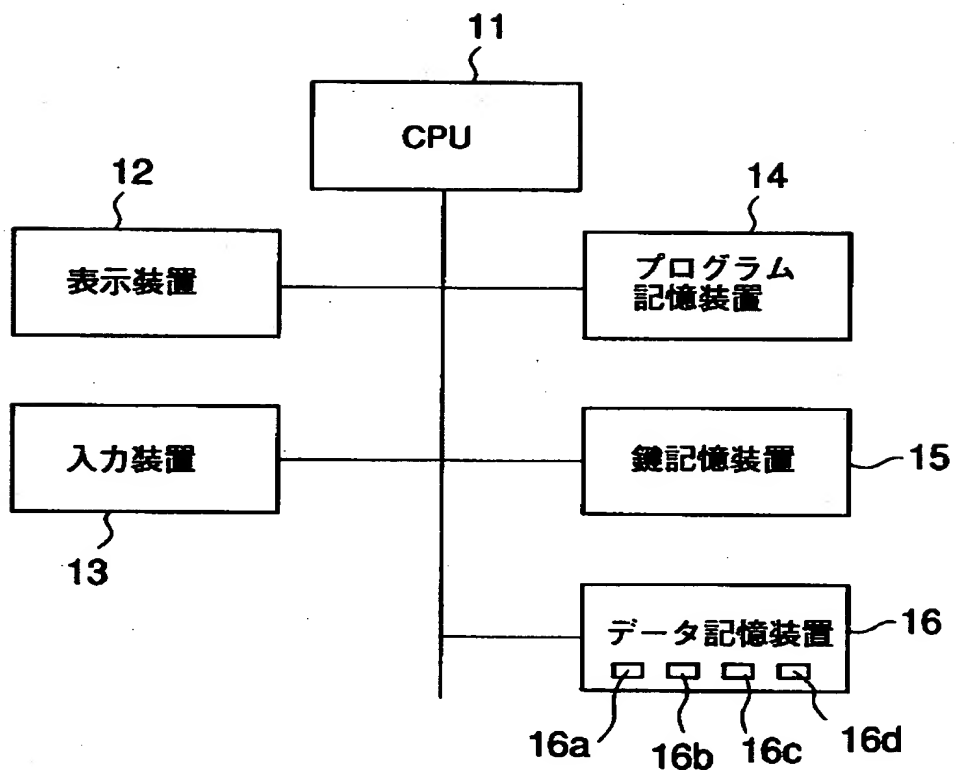
3 1 …検索依頼装置

3 2 …復号化装置

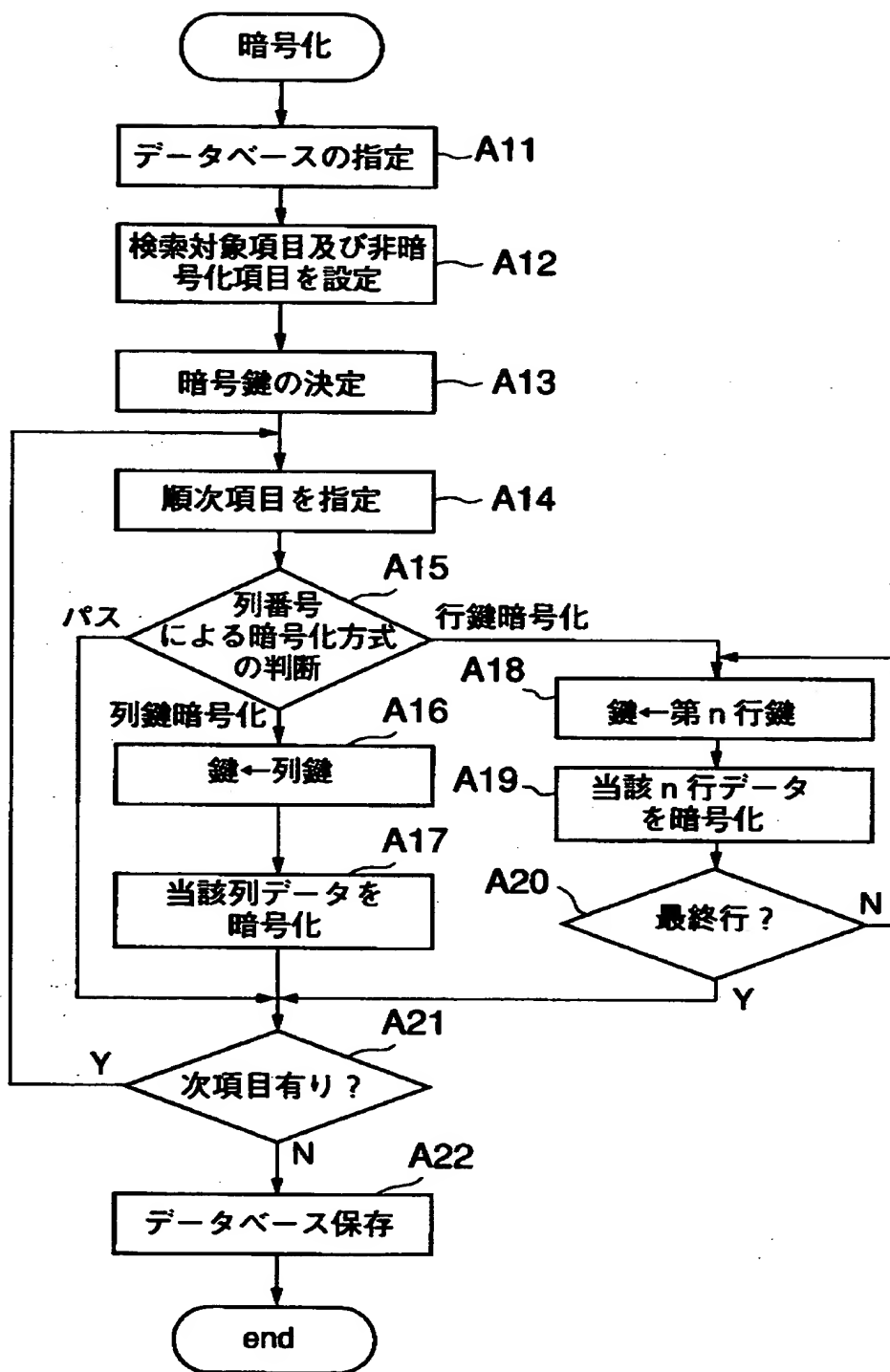
【書類名】

図面

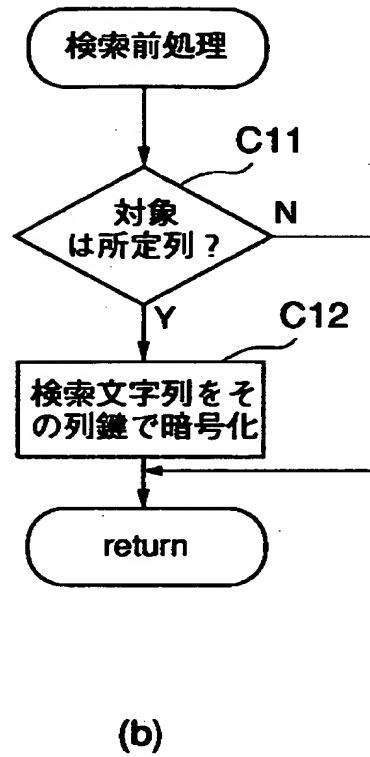
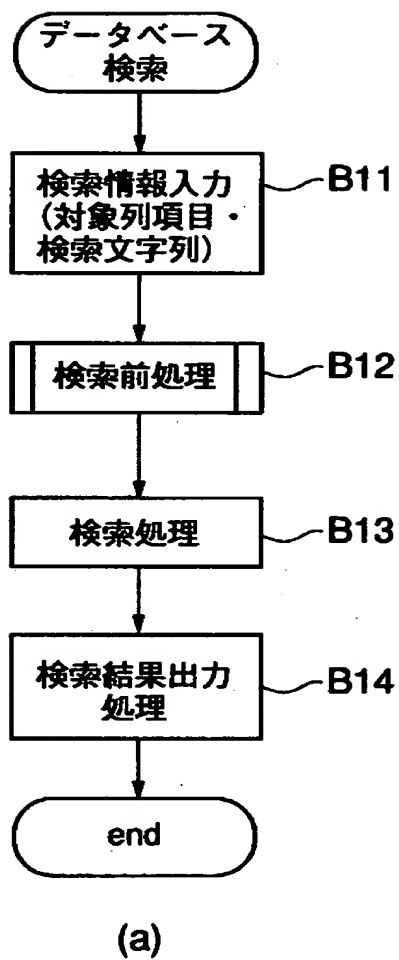
【図 1】



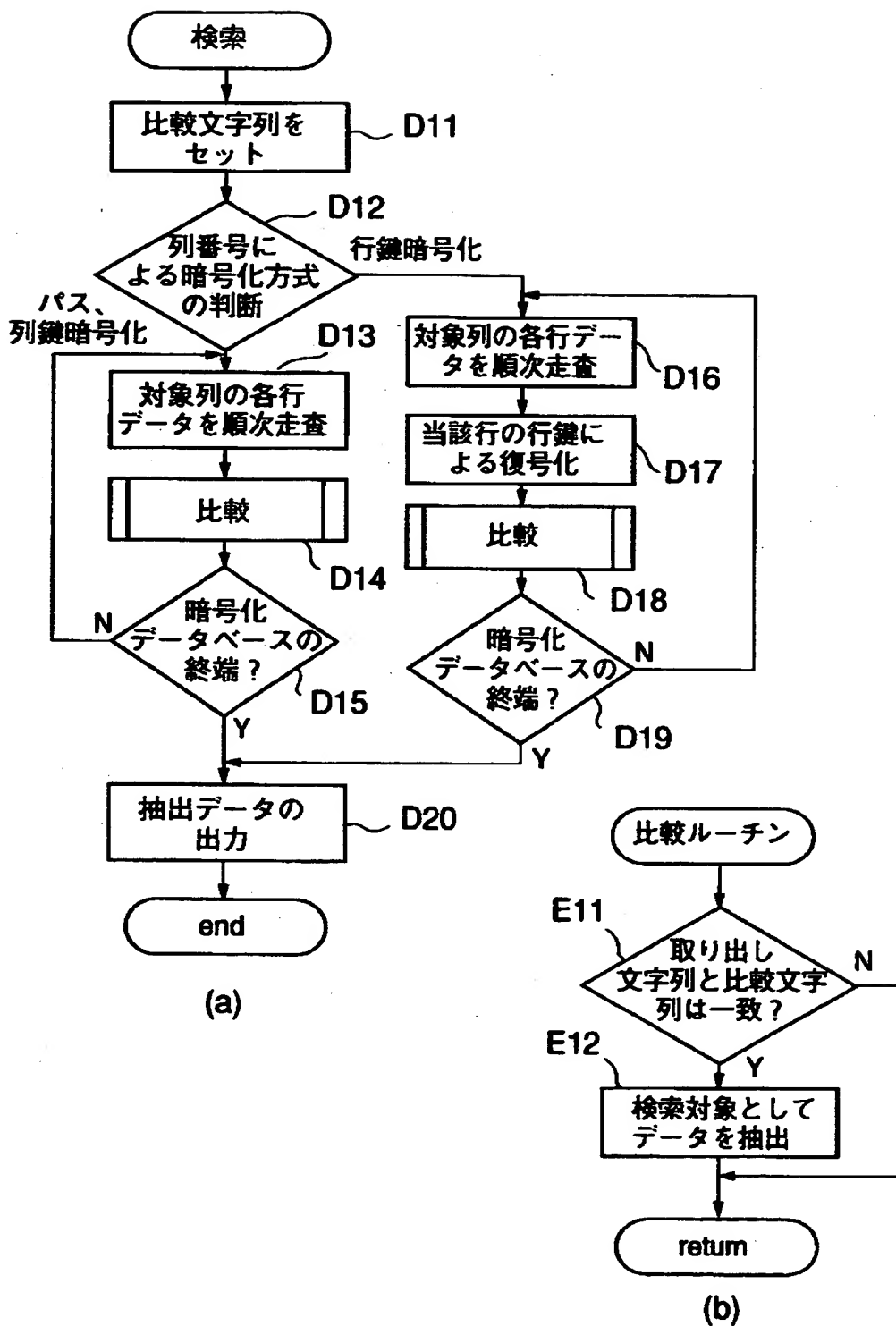
【図 2】



【図 3】



【図 4】



【図 5】

(a)

number	name	state	weight	height	age	phone
1	Jhon	New York	63	130	22	407-228-6611
2	Chris	Florida	72	190	21	123-456-7890
3	Michael	Minnesota	65	163	27	101-202-3030
4	David	Iowa	63	145	34	523-761-0045
5	Mark	New York	65	152	30	832-962-9001
6	Daniel	Iowa	68	170	25	231-981-9454
7	George	Idaho	69	180	31	561-545-4389
8	Henry	Florida	71	165	22	239-203-9800
9	Joe	New Jersey	66	163	27	239-129-9898

暗号化 列鍵  
行鍵

(b)

number	name	state	weight	height	age	phone
1	wJls	noevjlc	qw	ywe	jh	igdlitythDSk
2	ddGGa	h*/fDD	lr	Erw	hg	LKtYfDSkoKow
3	1jkl+P	gahf6xpVd	RK	Tyj	tY	hkllydageQk
4	3eK@s	kHHS	kd	DHH	Kl	d+fDIKnBerJf
5	eriN	noevjlc	jD	iOO	Gv	wsdERfvW2Sdf
6	F>sSlu	kHHS	8u	lki	ij	1xcVImFmkjpo
7	{:ld?k	IJHFD	HH	lpa	LK	kjwDkJGvfDoa
8	rhJKd	h*/fDD	ew	Aij	jh	e419h-ka+qwh
9	ifd	ASoChijIO-	Df	lky	tY	qLFUicVkj@kI

復号化 列鍵  
行鍵

(c)

number	name	state	weight	height	age	phone
1	Jhon	New York	63	130	22	407-228-6611
2	Chris	Florida	72	190	21	123-456-7890
3	Michael	Minnesota	65	163	27	101-202-3030
4	David	Iowa	63	145	34	523-761-0045
5	Mark	New York	65	152	30	832-962-9001
6	Daniel	Iowa	68	170	25	231-981-9454
7	George	Idaho	69	180	31	561-545-4389
8	Henry	Florida	71	165	22	239-203-9800
9	Joe	New Jersey	66	163	27	239-129-9898

【図 6】

列鍵

number	name	state	weight	height	age	phone
無し	"apple"	"orange"	行鍵	行鍵	"lemon"	行鍵

行鍵

number	
1	"tiger"
2	"dog"
3	"cat"
4	"mouse"
5	"elephant"
6	"cow"
7	"pig"
8	"rabbit"
9	"lion"



【図 7】

(a)

number	name	state	weight	height	age	phone
1	Jhon	New York	63	130	22	407-228-6611
2	Chris	Florida	72	190	21	123-456-7890
3	Michael	Minnesota	65	163	27	101-202-3030
4	David	Iowa	63	145	34	523-761-0045
5	Mark	New York	65	152	30	832-962-9001
6	Daniel	Iowa	68	170	25	231-981-9454
7	George	Idaho	69	180	31	561-545-4389
8	Henry	Florida	71	165	22	239-203-9800
9	Joe	New Jersey	66	163	27	239-129-9898

暗号化 合成鍵

(b)

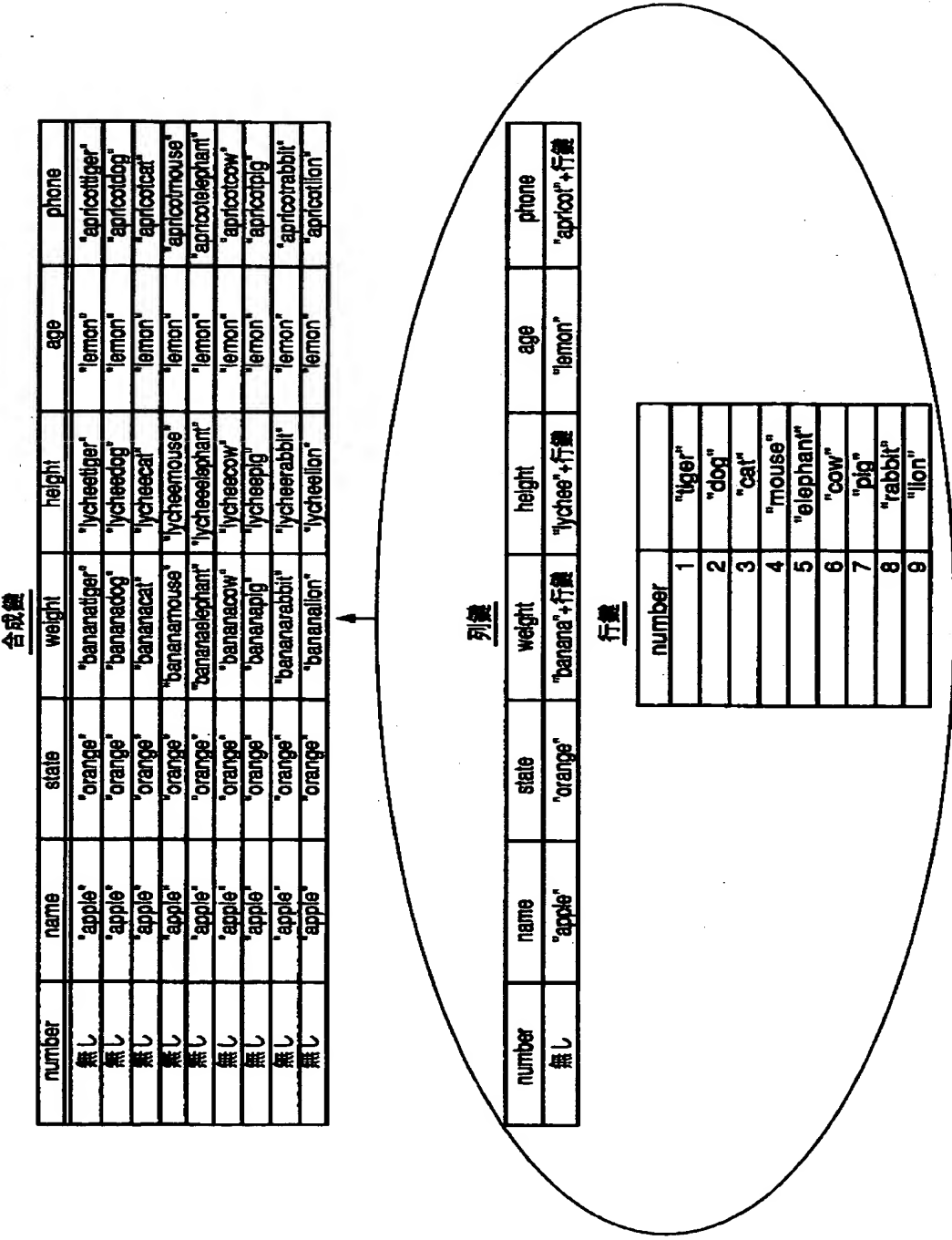
number	name	state	weight	height	age	phone
1	wJls	noevjlc	xo	qwe	jh	dfghaJ;lkqlu
2	ddGGa	h*/fDD	wi	kIA	hg	qwTyIBnDFIKj
3	1jkI+P	gah{6xpVd	hj	IKJ	tY	DafgiqlkimD—
4	3eK@s	kHHS	s?	SGA	KI	hi"khaTyqfXd
5	erIN	noevjlc	d-	ASD	Gv	8uyDBmAkolka
6	f>sSlu	kHHS	I*	qoK	ij	jhtvbnMKJASW
7	};ld?k	IJHFD	df	sLL	LK	IQwSRyuiokjq
8	rhJKd	h*/fDD	Ws	tyH	of	Dfha*kagil
9	ifd	ASoChijIO-	qo	H2a	Ga	lkjHYAGoiuq

復号化 合成鍵

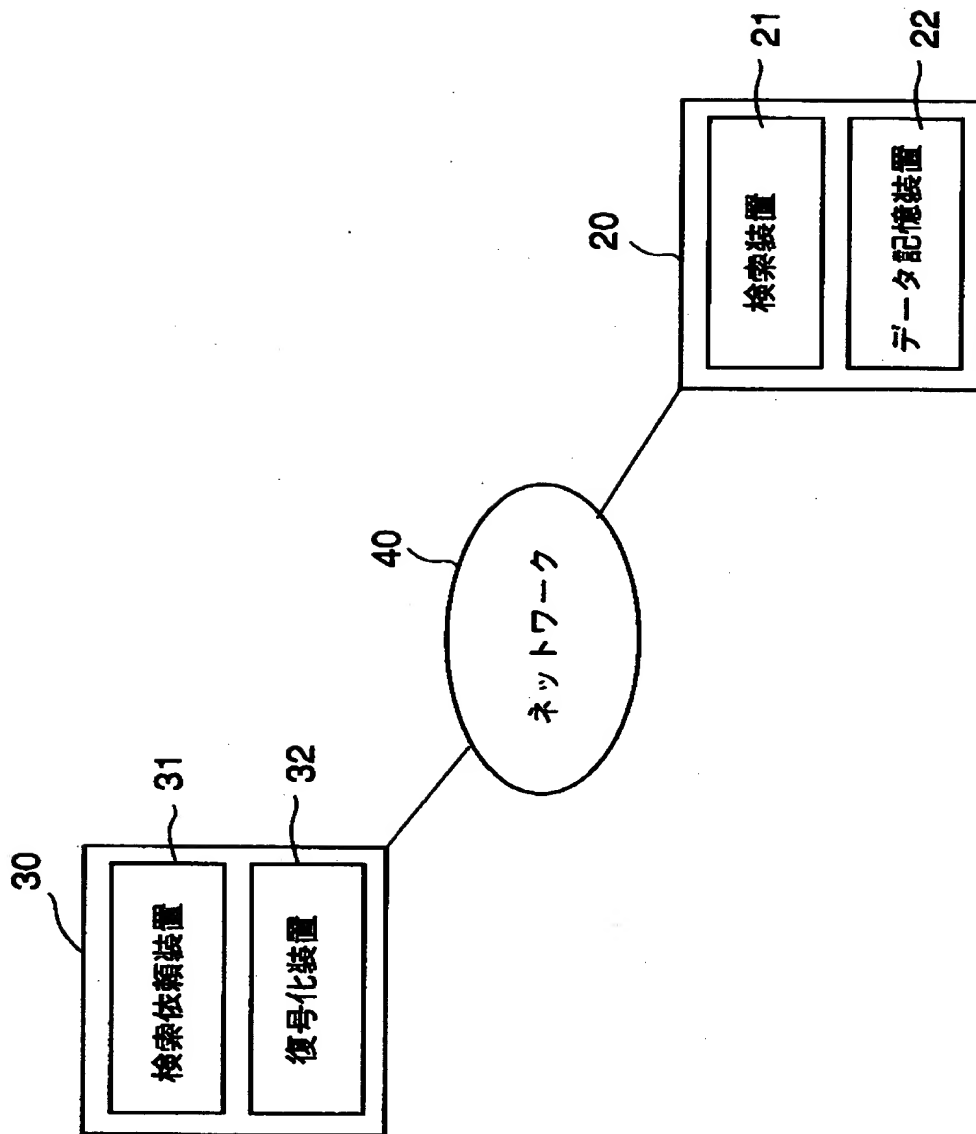
(c)

number	name	state	weight	height	age	phone
1	Jhon	New York	63	130	22	407-228-6611
2	Chris	Florida	72	190	21	123-456-7890
3	Michael	Minnesota	65	163	27	101-202-3030
4	David	Iowa	63	145	34	523-761-0045
5	Mark	New York	65	152	30	832-962-9001
6	Daniel	Iowa	68	170	25	231-981-9454
7	George	Idaho	69	180	31	561-545-4389
8	Henry	Florida	71	165	22	239-203-9800
9	Joe	New Jersey	66	163	27	239-129-9898

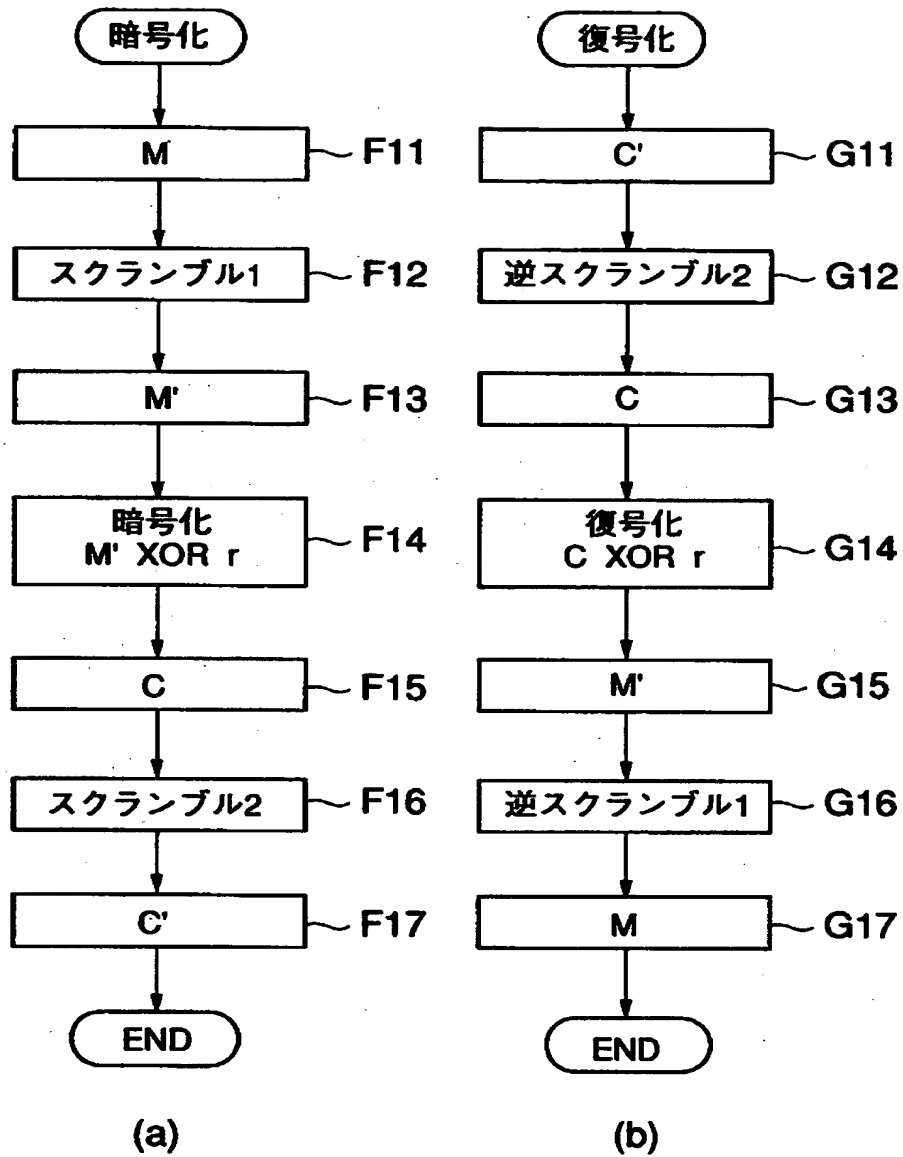
【図 8】



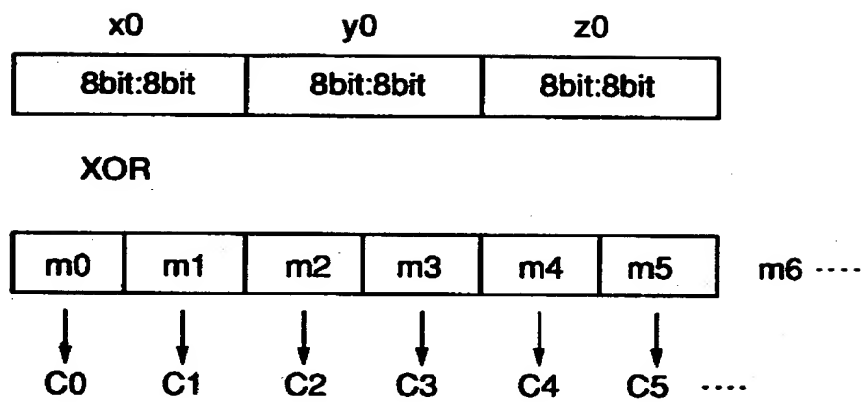
【図 9】



【図 10】



【図 1 1】



【書類名】 要約書

【要約】

【課題】 データベースのセキュリティを確保すると共に、検索に利用される項目については高速検索を可能とする。

【解決手段】 データベースを暗号化する際に、検索に利用される列項目のデータについては鍵記憶装置 1 5 に記憶された当該列項目に共通の列鍵を用いて暗号化し、その他の列項目のデータについては鍵記憶装置 1 5 に記憶された各行毎に固有の行鍵を用いて暗号化してデータ記憶装置 1 6 のデータベース格納エリア 1 6 a に保存しておく。このように、各行毎に鍵を異ならせることでセキュリティを高めることができ、検索時には、検索用として入力されたデータを前記所定の列項目に共通の列鍵を用いて暗号化し、その暗号化された検索用データとデータベース格納エリア 1 6 a 内の暗号化データベースとを比較することで高速検索を実現できる。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号                    [ 0 0 0 0 0 1 4 4 3 ]

1. 変更年月日            1 9 9 8 年   1 月   9 日

    [変更理由]            住所変更

        住 所            東京都渋谷区本町 1 丁目 6 番 2 号

        氏 名            カシオ計算機株式会社